Pricing QoS in Internetworks

Safiullah Faizullah
Computer Science
Rutgers University
Piscataway, NJ 08854, USA
sfaizull@paul.rutgers.edu

Ivan Marsic
Electrical and Computer Engineering
Rutgers University
Piscataway, NJ 08854, USA
marsic@caip.rutgers.edu

Abstract

Ever since its inception, the Internet has seen unprecedented growth. Consequently, researchers have been actively looking for means and ways to influence the behavior of its selfish users. Pricing was soon realized as the regulatory tool to provide proper incentives so that users' self-interest will lead them to modify their usage according to their needs. This leads to better overall network utilization and enhanced users' satisfaction. In this work, a scalable pricing framework for QoS capable networks supporting real time, adjustable real time, and non-real time traffic is studied. The scheme, which belongs to usage-based methods, is independent of the underlying network and the mechanisms for QoS provisioning. The framework is credit-based ensuring the fairness, comprehensibility, and predictability of usage cost. On the other hand, it provides means for the network providers to ensure, with high probability, cost recovery and profit, competitiveness of prices, and encouragement of client behaviors that will enhance the network's efficiency. This is achieved by appropriate charging mechanisms and suitable incentives. The implementation and usage costs of the framework are low. Simulation results suggest that users have better overall satisfaction; better network utilization is achieved while reduced call blocking probability is observed.

Keywords- Quality of Service, Pricing, Utilization, Call Blocking Probability, Users Satisfaction, Congestion Control

1. Introduction and Related Work

There has been very limited work in pricing. On top of this, the rapidly changing Internet characteristics (slowly transforming from best-effort to QoS capable) necessitates the need to devise new and improved pricing framework suitable for multi-services environment, where each service can support and guarantee QoS. This change is made possible with tremendous technological improvements, which resulted in better hardware, software and intelligent protocols, thus, reshaping the entire pricing research [0-1].

Traditionally, the individual users have not been charged for their use of networks, and have not generally been aware of the impact of their use on network performance. As a result Internet users have increased substantially with unfriendly and selfish attitude. The phasing out of Federal government funding of Internet operation in the United States has necessitated some form of alternative funding, such as revenue from fee for service operation. The traditional pricing is either free (subsidized through institutional funds) or flat-rate for unlimited usage. Some variations have been pricing bandwidth of the connection, or flat-rate to a certain hours and per hour charges thereafter [2, 3, 4-13, 15-16].

However, the greatly increased usages of the Internet and the resultant performance degradation have focused attention on the inefficiencies of the traditional pricing structures and their shortcomings. Also, a renewed emphasis on the research to improve hardware, software and protocols is needed particularly in the absence of proper incentives to act as congestion control. While there has been dramatic and outstanding success in the infrastructure research (resulting in high bandwidth backbones having gigabits transfer capability, widespread availability of PCs, easy network connections from homes, faster routers and sophisticated protocols), there has been a severe vacuum in pricing research. Since the traffic demands increase as the bandwidth (and other resources) improves, it is a mistake to argue that over-provisioning the capacity is the only solution for achieving high network performance [5]. Efficient pricing mechanisms coupled with traditional congestion control protocols are thought to be the ultimate solution to congestion control, which will result in better overall network performance. These pricing mechanisms are based on user incentives (particularly performance versus monetary as well as administrative) that seem to be the answer to the challenges posed by future Internet. An additional motivation for imposing a pricing scheme is to give users knowledge about the value of what they do to other users, and an interest to act so as to reduce harm done to others (social incentives).

It is expected that in the very near future integrated QoS capable networks (referred to as "NextInternet" in this paper) will emerge which provides a variety of transmission services, such as telephony, video, VoD, Interactive Games, Teleconferencing, and file transfer and all the other traditional Internet services. The system will be capable to negotiate QoS parameters and upon accepting user connection, it will be responsible for guaranteeing the agreed quality. Best-effort will be one of such services. Pricing is important but non-critical in today's Internet. In NextInternet the issue of pricing is more relevant than it is today. For otherwise, every user can and will opt for highest QoS available thus creating a huge congestion problem- thus the role of incentives. Also, it will result in high call admission blocking probability.

Note that there are great differences among the services offered by the QoS network; therefore, one might ask whether the prices of these service should also differ, and if so, how? Also, when more than one parameters are involved, for example av. delay = 1ms and BDW= 4Mbs versus av. delay = 2ms and BDW= 8Mbs or 1% packet loss probability and 400ms av. delay versus 2% packet loss probability and 200ms av. delay. How to price them? What ratio? Which should cost more? Integrating multiple services into a single network generates economies of scope, however heterogeneous services complicate pricing decisions.

There are a number of authors [1-5, 7] who have worked on the issue of how to price a network that offers heterogeneous services. A comprehensive survey of previous work [2-5, 6-13, 15-16] was presented in our previous work [1, 17].

The rest of the paper is organized as follows. After presenting the motivation in this section, the pricing scheme is presented in section 2. Section 3 presents charging methods with pricing agents explained in section 4. In section 5, utility functions are introduced. Simulation results are presented in section 6 and Conclusion outlined in section 7, which is followed by acknowledgements and references.

2. The Pricing Scheme

In this section, our proposed scheme is briefly explained. This scheme was introduced in [1, 17]. It is a simple method yet covers the most important aspects of a practical pricing framework. It is scalable framework for QoS capable internetwork (consisting of a collection of domains) supporting real time, adjustable real time, and non-real time traffic. We consider Bandwidth, Average Delay, Delay Jitter, and Packet Loss Probability as QoS parameters in this work. The scheme, belonging to usage-based [2, 3, 4] methods, is independent of the underlying network and the mechanism for QoS provisioning, and it can be deployed in any QoS capable environment where best-effort is one of the available classes. The scheme is credit-based ensuring the fairness (from users' point of view), comprehensibility, controllability, predictability, and stability. On the other hand, it provides means for the network providers to ensure cost and profit recovery, competitiveness of prices, and encouragement of client behaviors that will enhance the network's efficiency. This is achieved by appropriate charging mechanisms and suitable incentives.

In conventional QoS models, the required bandwidth or the BitRate has no rigid relation with the PacketRate. However all the forwarding takes place based on packets. Therefore we choose that PacketRate should be used instead of BitRate as the QoS Parameter. All other QoS parameters also consider a per packet behavior (e.g., Delay, Jitter and LossProbability). Hence Considering PacketRate instead of BitRate will help relate all the QoS Parameters.

The Packet Size is fixed for the constant bit rate service while it is variable for VBR service. The users/applications need to declare the mean packet size (and possibly the standard deviation, S.D., which would be helpful in QoS-Routing mechanism) in advance.

The main problem in pricing of Internetworks services is to find a metric that fairly represents the relative merit of each service. In this work, we present a technique that not only includes the definition of such a metric but also covers the practical issues of implementation in the existing or future Internetworks.

2.1. Quality Metric

Complicating the pricing of QoS capable network is the lack of a method to compare the quality of different types of services. Even comparing quality within the same type of service is difficult. For example two connections of same type of service but different hop count (or distances).

In other words, we need a ***common metric*** ***f***(TypeOfService, QoS) such that:

$$f(T1,Q1) > f(T2,Q2)$$

implies user of service T1 is more satisfied than user of serviceT2.

The common metric is a good indicator of the price, representing relative merit of each service. We term this common metric as Price (P(s_{QoS})), where s_{QoS} is the set of QoS parameters (Bandwidth, Av. Delay, Delay Jitter, Packet Loss probability, etc.). P(s_{QoS}), is also used as basis for charging.

2.2. Types Of Service

We categorize the services as the following broad Type of Services classes, each having a different set of QoS parameters. Within each type of service different Quality of Services are available by adjusting the QoS parameters [1]. Motivation for this categorization and detailed treatment is provided in [1].

1) Real Time Service (RT):
Services that have critical/tight upper bound on the time at which the bits are arrived. Discarding any data that are arrived beyond ETA. Packets delayed are of no use to the users. Examples of such services are Telephony, Teleconferencing and covering ATM's CBR and rt-VBR.

2) Adjustable Real Time Service (A-RT):
In this service class, we do not discard data if it is delayed. Instead, the ETA is adjusted as long as the occurrences and durations of these delays and discontinuations are within some acceptable bounds. For example, half-duplex video may be resumed after a little pause due to the delay in packets arrival. In this case we can adjust the Acceptable Delay parameter to be increased by the stalled time. Examples of such services are Video on Demand, Video, Interactive Games, and Distance Learning.

3) Non Real Time Service:
Service, without QoS guarantees, which exploits available resources. Examples of such service are traditional email and File Transfer.

2.3. Pricing for RT Service

In this section, we derive the metric Price for Real Time Service class. In order to derive such a metric, for pricing purposes, we reduce the QoS parameters set through a series of redefinition of these parameters such that P(s_{QoS})) involves as few parameters as possible. At the end we are getting a raw indicators, P(s_{QoS})), which satisfies the property of common metric. Table 1 shows the treatment of QoS parameters where "=" means the particular parameter is used as-is in the derivation of P(s_{QoS})), and "→" means that the parameter at right-hand side is transformed into the parameter in left-hand side. The parameters (1)-(4) are only used to derive P(s_{QoS})).

<u>QoS Parameters:</u>	<u>Translates to:</u>	
PacketDelay	= Delay	(1)
DelayJitter	→ Delay	
PacketRate	= PacketRate	(2)
PacketLossProbability	→ PacketRate	
MeanPacketSize	= PacketSize	(3)
Distance (Avg. Hop Count)	= Distance	(4)

Table 1

Delay Jitter → Delay:
Jitter is removed by the use of play-out buffers, which introduces increase in the end-to-end Delay.

In other words, Delay Jitter is reduced into an increase in the end-to-end Delay as follows:

Revised Delay = Acceptable Delay +Acceptable Delay Jitter.

PacketLossProbability → PacketRate:
For pricing purposes again, we absorb Packet Loss Probability (PLP) into revised Packet Rate, shown via the following example:

<u>Connection 'A'</u>	<u>Connection 'B'</u>
PacketRate = 20 P/Sec.	PacketRate = 18 P/Sec.
PLP = 10% (i.e. 2 P/Sec)	PLP = 0%
	BestEffort = 2 P/Sec.

Revised PacketRate = Requested PacketRate*(1-PLP).

<u>PacketRate:</u>
We charge for any packets that are received in time at the destination. Hence higher PacketRate connections will end up paying more. Call Admission mechanisms should accept only those PacketRates that can be accommodated. Policer doesn't let the sender, using leaky bucket, to send at higher than agreed rates.

<u>Packet Size:</u>
Being usage-based scheme, larger Packet Size means more usage of resources and, therefore, more cost. (i.e., larger

PacketSize => more cost). Price is directly proportional to the Packet Size.

Delay:
Since delay has two components, namely Queuing Delay and Propagation Delay (in other words Delay = Queuing Delay + Propagation Delay), by using Standardized Propagation Delay (SPD) with some cushion to cope with larger routes, we define Acceptable Queuing Delay as:

AcceptableQueuingDelay = Delay – SPD.

Requirement of lesser AcceptableQueuingDelay (AQD) means more cost and hence Price is inversely proportional to the AcceptableQueuingDelay.

Distance:
To provide same QoS, connections between distantly located hosts need more resources than those between closely located hosts. Again using Standardized Distances (in Hops), the network Routing mechanism should find smaller routes to save costs. Here, Price is directly proportional to the Distance.

Per-Packet Accounting:
We compute ETA for each packet based on the Acceptable PacketDelay. Any Packet that has arrived within its ETA, satisfies the required QoS and is charged at:

Price = ActualPacketSize * Distance / AQD.

Any Packet slot that is missed by the sender is charged at:

Price = MinimumPacketSize * Distance / AQD.

The Packets that are delayed beyond their ETA are credited at:

Price = ActualPacketSize * Distance / AQD.

The packets that are dropped due to congestion are credited at:

Price = MeanPacketSize * Distance / AQD.

2.4. Pricing of Adjustable RT Service

Recall that A-RT Service does not discard data if it is delayed. Instead, each acceptable pause results in an increase in Acceptable Delay parameter by the stalled time and therefore adjusting ETA. In case the frequencies or durations of such delays are violating the agreement, users will be credited using the same method of RT service class.

Noting this difference, we are using the same formulae of RT service with adjusted Acceptable Delay being recalculated after each acceptable pause. This yields lower overall price for A-RT service class.

3. Charging Methods

A number of charging related issues are discussed in this section. In [1], we provided detailed treatment of this issue. The following is a generic charging formula for usage-based pricing where the network administrators set different coefficients. The formula consists of three components namely: usage charges, reservation charges, and access charges.

$C_{\text{traffictype}}(\sigma_{\text{QoS}}) = \alpha_{\text{traffictype}} * P(\sigma_{\text{QoS}}) + \beta * R(\sigma_{\text{QoS}}) + \gamma;$

Where, $C_{\text{traffictype}}(\sigma_{\text{QoS}})$: Cost for traffictype (e.g. RT, A-RT, etc.).
$\mathbf{P}(s_{\mathbf{QoS}})$: Price calculated by Pricing mechanism.
$\mathbf{R}(s_{\mathbf{QoS}})$: Resource reservation charge (may include connection establishment charge).
$s_{\mathbf{QoS}}$: QoS parameter set (Bandwidth, Packet Loss Probability, Average Delay, Delay Jitter, …)
$a_{\mathbf{traffictype}}$: Coefficient for usage charges.
b: Coefficient for reservation.
g: Fixed access charge.

In this work, since our pricing framework is independent of underlying network infrastructure, we can only apply a fixed charge for connection establishment. This can be justified as resource consumption is captured in usage charges indirectly.

Also, we assume that the access charge component is not per connection rather the access providing entity can charge this per month and during the revenue distribution this amount is charged by the billing mechanism.

Therefore, the formulae for different type of services that we consider in this work are:

$\mathbf{C}_{\mathbf{rt/art}}(s_{\mathbf{QoS}}) = \alpha_{\text{rt/art}} * P(\sigma_{\text{QoS}}) + R$ and
$\mathbf{C}_{\mathbf{be}}(s_{\mathbf{QoS}}) = \alpha_{\text{be}} * P(\sigma_{\text{QoS}});$

Where, R is a fixed connection establishment charge.

Note that it makes sense to assume that α_{rt} is higher in value than α_{art} with α_{be} being the lowest of all three values. We assume these coefficients are defined by pricing agents- discussed below- and not individual routers that are involved in the connection. Network providers can

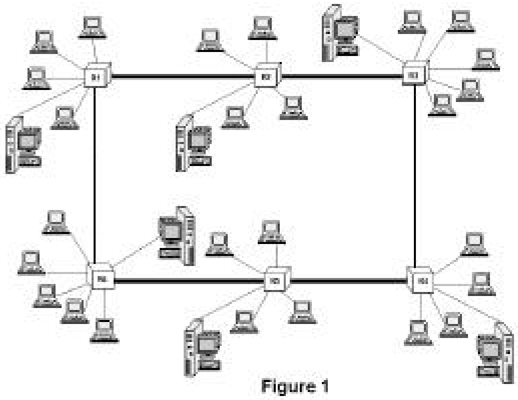

Figure 1

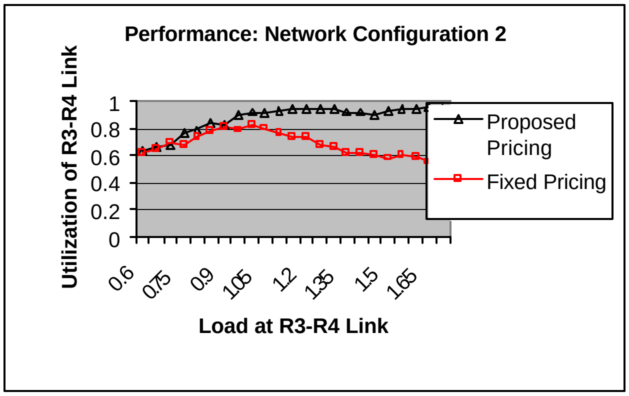

Figure 2

make these coefficients sensitive to different time periods. However, since network performance degradation may become apparent in extreme cases, the swing should be carefully designed.

Some other aspect of charging mechanism is omitted here and can be found in [1].

4. Pricing Agents

In order for our Pricing Framework to be scalable, we choose endpoints (edge routers) to be the place for accounting, where dedicated agents receive duplicate headers from corresponding edge routers on the receiving hosts side. They are only dummy hosts with no overhead to the network. Pricing Agents discard the messages after logging the header information. In addition, they can be used to perform other activities such as coefficients estimations, providing charging information, acting as call admission mechanism, deciding queue sizes for network providers, and a host of other activities including Metering, Billing, Advertisements and Revenue Distribution. It can also bill electronically and receive payments via network too.

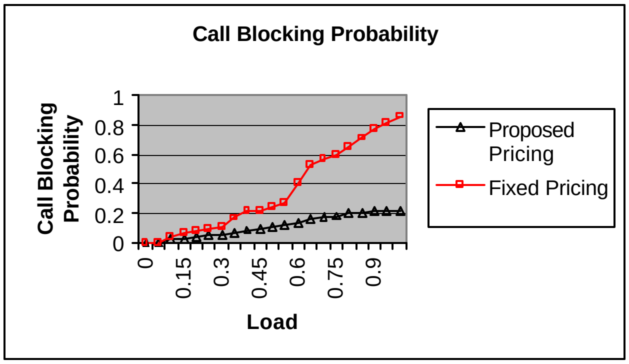

Figure 3

The Pricing Framework is implemented in a complete decentralized manner. A handshake is needed between the Call Admission mechanism of the network and the Pricing Mechanism so that the later knows about the QoS parameters (and their values) that the network guaranteed to the user. Also, users can enquire about their accounts. In case the QoS network supports renegotiations [14], this handshake needs to take place every time such renegotiation is accepted by the network- so that the price and finally net charge is calculated accordingly. Pricing Agent can also be used to implement a renegotiation mechanism such as the one reported in [14].

5. Utility Functions

We base the definition of the utility functions used in this work on the economic theory, which states that given a congested resource, the price one pays to send a message (i.e., its utility) should reflect the loss of utility inflicted on the other users whose messages did not get the same treatment. The utility functions need to show the performance one’s application gets and the price they pay for.

In general the utility function takes the following form:

$$\mathbf{U}(\mathbf{P}(\sigma_{QoS})) = \mathbf{V}_{\mathbf{traffictype}}(P(\sigma_{QoS})) - \mathbf{C}_{\mathbf{traffictype}}(\sigma_{QoS}).$$

Where $V_{\mathbf{traffictype}}(P(\sigma_{QoS}))$ is the apparent degradation (in favor of other users) to a user while $C_{\mathbf{traffictype}}(\sigma_{QoS})$ is the price one is charged for. $U(P(\sigma_{QoS})) \rightarrow +R$ is a mapping from a non-negative real number to non-positive real number showing the worth utility perceived by a user (for a usage request with quality given by σ_{QoS} and for which cost $C_{\mathbf{traffictype}}(\sigma_{QoS})$ was paid).

Following is one set of $V_{\mathbf{traffictype}}(P(\sigma_{QoS}))$ for real-time, adjustable real-time, and non-real-time traffic.

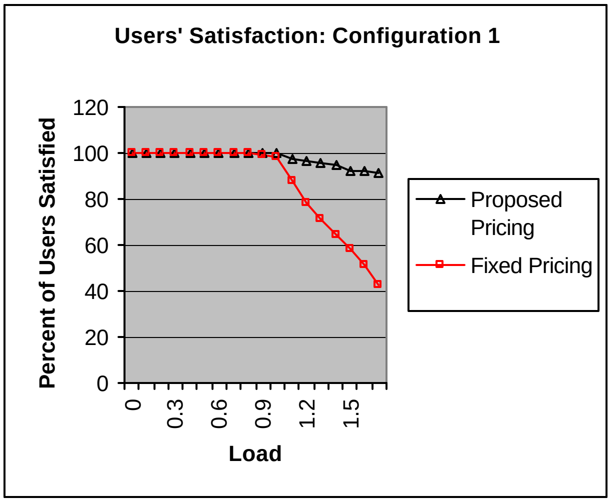

Figure 4

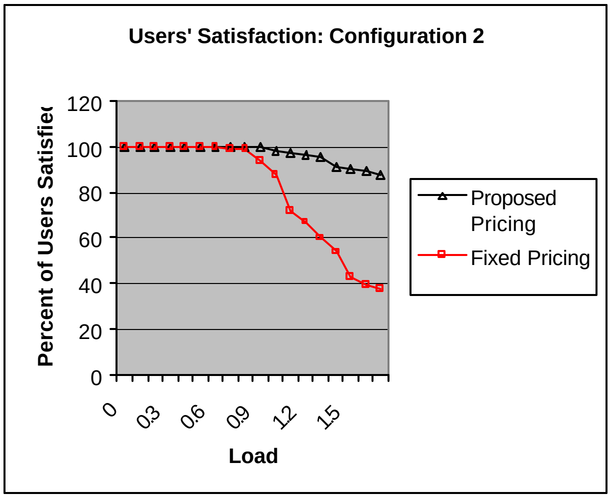

Figure 5

$V_{rt}(P(s_{QoS}))$ = -number of packets that did not meet ETA.
$V_{art}(P(s_{QoS}))$ = -number of packets that did not meet adjustable ETA.
$V_{nrt}(P(s_{QoS}))$ = -number of packets that did not arrive within a loose bound (generally few minutes).

User satisfaction can be asserted by using utility functions, given above, for different traffic types. Utility functions defined in this section can also be used to compare different pricing schemes (for example to compare flat-rate versus differentiated pricing).

6. Simulations Results

Users satisfaction, expressed using the utility functions, network utilization and call blocking probability are studied through a series of simulations conducted on two network configurations. The first configuration consists of one bottleneck link connecting two routers, where all one-way communication is from hosts attached to one router to hosts attached to the opposite router only. The second configuration is given in Figure 1, which consists of six bottleneck links. Bottleneck links connect two routers and are 2 Mbps with 10 ms of propagation delay. All others links, which connect hosts to a routers are 10Mbps with propagation delay of 1 ms.

We used charging coefficients (i.e., α_{art}) of 4.164×10^{-4} (this amounts to $0.2/min for 64Kbps bandwidth) for A-RT traffic while $\alpha_{rt} = 8.328 \times 10^{-4}$ (amounting to $0.4/min for 64Kbps bandwidth) for RT traffic is used in our experiments. We used $\alpha_{be} = 0$ for best-effort traffic, same as present day Internet. Fixed connection establishment cost (i.e., R) was also kept at 0. We used 1024 Bytes packet size. Also, a request for call admission was accepted if network utilization was under 95% and a path with required bandwidth was found. No renegotiations were allowed in these experiments. During the simulation lifetime there were between 24 to 48 flows including some 4-8 background flows (simulating best-effort traffic). User requests arrived according to Poisson distribution with a rate of a request per μ minutes, and the duration of connections exponentially distributed with a mean λ. User traffic was generated using On-Off model with the On and Off periods being exponentially distributed (with means λ_{on} and λ_{off}). We used a variety of traffic mix in our simulations. Users chose between RT and A-RT services classes. This was modeled as a bimodal distribution with a fraction τ of users choosing RT and $1-\tau$ opting for A-RT. We assumed a single domain network.

Here, offered load (load for short) is defined as ratio between the total amount of bandwidth reserved (for all connections) and total bandwidth of bottleneck. Also, link utilization is defined as ratio between the total amounts of bandwidth required (for all connections) and total bandwidth of bottleneck. In our first experiment, the single bottleneck link was observed while in the second experiment one of the bottleneck links (in this case R3-R4, Figure 1) was studied. Experiments were conducted for about two hours and results were observed after the lapse of an initial warm-up time of few minutes- until link utilization reached 60%. Data were collected at different load levels.

Figure 2 shows utilization as a function of load for network configuration given in Figure 1. Utilization increases continuously when pricing is used and settles near the target utilization for the experiments. On the other hand, utilization is low (with various degrees) when pricing is not used as a means to provide monetary incentives for the users. In this case, users opt for the highest quality of service each time they request a connection establishment. The end result is not only low utilization but also high blocking probability for arriving requests; this is shown in Figure 3.

Figure 4 shows users' satisfaction as a function of load for the first configuration (where the single bottleneck link network was studied). In Figure 5, users' satisfaction is given for the second configuration shown in Figure 1. Both of these figures show that users' satisfaction deteriorates (normalized for all user), as we increase the offered load, in case pricing is not employed. This is because packet drops as network becomes congested (due to high load). This combined with apparent increase in call blocking probability, given in Figure 3, completes the picture of deteriorated users' satisfaction in case they do get connection as well as when their requests for connections are refused. Users satisfaction was determined by using the functions defined in section 5.

As a direct conclusion of these observations, we can deduce that users have better overall satisfaction. Better network utilization is achieved while reduced call blocking probability is observed.

7. Conclusion

A computationally simple and scalable pricing framework was proposed [1, 17] and is studied in this work. Further details can be found in the extended version of this paper in [17]. Better network utilization, lower call blocking probabilities, and better overall users satisfaction are some of the direct results (based on analysis of simulation results) of employing pricing in QoS capable networks.

As mentioned earlier all the pricing related work is done either at edge routers or at dummy hosts attached to edges routers (called Pricing Agents) and therefore it is easy to scale.

8. Acknowledgements

The authors would like to thank Mr. Arshad Shaikh of Computer Science Department, Rutgers University for discussions and useful comments. We would also like to thank Nasir Qamar for his help and encouragements.

9. References

[0] Safiullah Faizullah and Ivan Marsic, "Pricing QoS: Simulation and Analysis", to appear in IEEE INT SS 01, Seattle, Washington, April 22-26, 2001.

[1] Safiullah Faizullah and Ivan Marsic, "A Pricing Framework For QoS Capable Internet", in PDCS 2000, Las Vegas, Nevada, Nov. 6-9, 2000, pp. 569-577.

[2] R. Cocchi, S. Shenker, D. Estrin and L. Zhang, "Pricing in Computer Networks: Motivation, Formulation, and Example," *IEEE/ACM Transactions on Networking*, vol. 1, no. 6, Dec. 1993.

[3] Colin Parris, Srinivasan Keshav and Domenico Ferrari, "A Framework for the Study of Pricing in Integrated Networks," Institute of Computer Science, Berkeley, California, no. TR-92-016, Mar. 1992.

[4] Domenico Ferrari and Luca Delgrossi, "Charging For QoS," in *Proc. of IwQOS*, (Napa, California), May 1998.

[5] Peter C. Fishburn and Andrew M. Odlyzko, "Dynamic behavior of differential pricing and Quality of Service options for the Internet," AT\&T Labs - Research, Florham Park, New Jersey, Jun. 1998.

[6] Qiong Wang, Jon M. Peha and Marvin A. Sirbu, "The Design of an Optimal Pricing Scheme for ATM Integrated-Services Networks," in *Proc. of MIT Workshop on Internet Economics*, Mar. 1995.

[7] Yannis Korilis and Ariel Orda, "Incentive Compatible Pricing Strategies for QoS Routing," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, (New York), Mar. 1999.

[8] Shaogang Chen and Kihong Park, "An Architecture for Noncooperative QoS Provision in Many-Switch Systems," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, (NY), Mar. 1999.

[9] A. Orda, R. Rom, and N. Shimkin. Competitive routing in multiuser communication networks. IEEE/ACM Trans. Networking, 1(5):510-521, 1993.

[10] Scott Shenker. Making greed work in networks: a game-theoretic analysis of switch service disciplines. In Proc. ACM SIGCOMM '94, pages 47-57, 1994.

[11] J. Hazlewood, Optimum Pricing as Applied to Telephone Ser., Review of Econo. Stud. 12,2 ('51), 67-78.

[12] J. MacKie-Mason and H. Varian, "Pricing the Internet," in *Public Access to the Internet*, Kahin, Brian and Keller, James, ed., Boston, Massachusetts, May 1993.

[13] Alok Gupta, Dale O. Stahl and Andrew B. Whinston, "Priority pricing of integrated services networks," in *Internet Economics*, Lee W. McKnight and Joseph P. Bailey, ed., pp. 323--352, 1997.

[14] Safiullah Faizullah and Arshad Shaikh, "A Feedback-Based UPC-Parameters Renegotiation Strategy", Proc. Of the Internl Conf. On Parallel and Distributed Processing Techniques and Application (PDPTA '99), Las Vegas, Nevada, Jun 28-July 1, 1999, pp. 937-943.

[15] B. Parenteau and N. Rishe, "Internet Pricing and Prioritization," in *Proc. of 4th International Workshop on Community Networking*, (Atlanta, Georgia), Sep. 1997.

[16] A. Viswanathan, N. Feldman, Z. Wang and R. Callon, "Evolution of Multiprotocol Labe Switching", IEEE Comm. Magazine, May 1998, pp. 165-173.

[17] Safiullah Faizullah, "A Pricing Framework For QoS Capable Internet", <http://paul.rutgers.edu/~sfaizull>, May 2000.